

# Introducing One Step Back Iterative Approach to Solve Linear and Non Linear Fixed Point Problem

Dohy Hong  
Alcatel-Lucent Bell Labs  
Route de Villejust  
91620 Nozay, France

dohy.hong@alcatel-lucent.com

## ABSTRACT

In this paper, we introduce a new iterative method which we call one step back approach: the main idea is to anticipate the consequence of the iterative computation per coordinate and to optimize on the choice of the sequence of the coordinates on which the iterative update computations are done. The method requires the increase of the size of the state vectors and one iteration step loss from the initial vector. We illustrate the approach in linear and non linear iterative equations.

## Categories and Subject Descriptors

G.1.0 [Mathematics of Computing]: Numerical Analysis—*Numerical algorithms*; G.1.4 [Mathematics of Computing]: Numerical Analysis—*Iterative methods*

## General Terms

Algorithms, Performance

## Keywords

Numerical computation, iterative method, fixed point

## 1. INTRODUCTION

Iterative methods to solve large sparse systems have been gaining interests in very different research areas and a large number of approaches have been studied: starting from Jacobi or Gauss-Seidel iteration methods, more recent works focuses on relaxation methods, Krylov subspace methods, the use of preconditioners, matrix decomposition, parallel computation etc. to solve linear or non linear problems [14], [8], [1], [7], [15], [6], [3], [16] to cite few of them.

In this paper, we propose a new iterative algorithm based on the anticipated consequence of the iteration at each coordinate level: we believe that this approach may bring significant improvement in a large class of linear and non linear problems and it seems that such an approach have not yet been studied by the research community. More precisely, we study the computation of the fixed point  $X \in \mathbb{R}^N$  solving

$$X = \mathbf{H}(X)$$

starting from the initial condition  $X_0 \in \mathbb{R}^N$ .

In linear algebra, this approach is equivalent to the idea of fluid diffusion that was called D-iteration in [11].

In this paper, we will not consider the convergence issue and assume that the Jacobi or the asynchronous version of

Gauss-Seidel style iteration of  $X_{n+1} = \mathbf{H}(X_n)$  converges to a unique fixed point (that's basically contracting operators, cf. for instance [2, 4, 5]).

In Section 2, we define the notations and the proposed One Step Back (OSB) algorithm. Section 3 illustrates two concrete examples to solve linear and non linear systems.

## 2. OSB ALGORITHM

We will use the following notations:

- $\mathbf{I} \in \mathbb{R}^{N \times N}$  the identity matrix;
- $\mathbf{J}_i$  the matrix with all entries equal to zero except for the  $i$ -th diagonal term:  $(\mathbf{J}_i)_{ii} = 1$ ;
- $\mathbf{H} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  an operator;
- $\mathcal{I} = \{i_1, \dots, i_n, \dots\}$  a sequence of the coordinates (or nodes),  $i_n \in \{1, \dots, N\}$ .

### 2.1 Iterative equations

The OSB iteration is defined by the triplet  $(\mathbf{H}, X_0, \mathcal{I})$  and exploits two state vectors of size  $N$ ,  $H_n$  (history) and  $F_n$  (residual fluid) based on the following iterative equations:

$$\begin{aligned} H_0 &= X_0 \\ H_n &= H_{n-1} + \mathbf{J}_{i_n}(F_{n-1}) \end{aligned} \quad (1)$$

and

$$\begin{aligned} F_0 &= \mathbf{H}(H_0) - H_0 \\ F_n &= (\mathbf{I} - \mathbf{J}_{i_n})(F_{n-1}) + \mathbf{H}(H_n) - \mathbf{H}(H_{n-1}). \end{aligned} \quad (2)$$

### 2.2 Sequence choice and distributed computation

The choice of the sequence  $\mathcal{I}$  depends on the computation costs structure that need to be optimized. One may consider those introduced in [12], [13] or [9] depending on the context.

For the distributed computation architecture, one may also use the one proposed in [10] for the linear case: the architecture for linear or non linear cases should be a priori the same and benefits from the asynchronous properties of the proposed method.

## 3. ILLUSTRATION

### 3.1 Linear equation: D-iteration

If we take  $\mathbf{H} = \mathbf{P}$  a linear operator, we have:

$$\begin{aligned} F_n &= (\mathbf{I} - \mathbf{J}_{i_n})(F_{n-1}) + \mathbf{H}(H_n) - \mathbf{H}(H_{n-1}) \\ &= (\mathbf{I} - \mathbf{J}_{i_n})F_{n-1} + \mathbf{P}\mathbf{J}_{i_n}F_{n-1}. \end{aligned}$$

Then we find back D-iteration equation cf. [11]. The computation gain for this case has been studied in details for instance in [12, 10, 13] and is skipped here.

### 3.2 Non linear equation: simple example

Consider the non-linear fixed point problem of dimension three:

$$\mathbf{H}(x, y, z) = (\sqrt{xy} + 1, (x + z)/4 + 1, (x + y)/4).$$

The fixed point of the above equation is  $((23 + \sqrt{379})/10, (23 + \sqrt{379})/30 + 16/15, (23 + \sqrt{379})/30 + 4/15) \sim (4.247, 2.482, 1.682)$ . The updates of the fluid vector  $F_n$  is based on the three increment functions:

$$\mathbf{H}(x', y, z) - \mathbf{H}(x, y, z) = (\sqrt{y}(\sqrt{x'} - \sqrt{x}), \frac{(x' - x)}{4}, \frac{(x' - x)}{4})$$

$$\mathbf{H}(x, y', z) - \mathbf{H}(x, y, z) = (\sqrt{x}(\sqrt{y'} - \sqrt{y}), 0, \frac{(y' - y)}{4})$$

$$\mathbf{H}(x, y, z') - \mathbf{H}(x, y, z) = (0, \frac{(z' - z)}{4}, 0).$$

Figure 1 shows the convergence of Jacobi, Gauss-Seidel and OSB iterations. One iteration is here defined as an update of three entries (with multiplicity for OSB) of the iterated vector. For OSB, we choose the  $n$ -th coordinate equal to the arg max of  $F_n$  in absolute value. The initial condition is set to  $(4.2, 1, 1.5)$ .

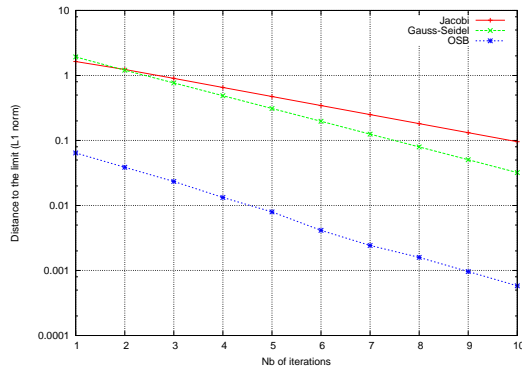


Figure 1: Illustration:  $N = 3$ .

Figure 1 illustrates well the advantage of optimizing the coordinate sequence order when the initial vector has some coordinates more closer to the limit than others: the gain factor compared to Jacobi iteration is here of two orders of magnitude at iteration 10! Note that in this case, starting from  $(0, 0, 0)$ , there is no big differences between those three methods (the matrix size is too small to observe significant differences).

This example is only shown for the sole purpose of the illustration of the potential impact and no theoretical guarantee on the convergence gain is given here. However, the author believes that we should have cases where the OSB method may provide substantial convergence improvement, since it can be applied to a very wide range of fixed point problems. The first factor to be considered to determine whether OSB method can improve the iterative computation cost is the complexity of the computation of the increment  $\mathbf{H}(H_n) - \mathbf{H}(H_{n-1})$ , which may simplify or introduce a cost overhead to Gauss-Seidel style *normal* iteration, and to

compare this complexity to the gain brought by coordinate level optimization.

Note finally that the one step iteration computation lost at the first iteration is recovered at the end considering  $H_n + F_n$  as the estimator (instead of  $H_n$ ).

## 4. CONCLUSION

In this paper, we described a new iterative method and illustrated its applications to two simple fixed point problems.

## 5. REFERENCES

- [1] W. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.
- [2] G. M. Baudet. Asynchronous iterative methods for multiprocessors. *J. ACM*, 25(2):226–244, Apr. 1978.
- [3] A. Berman and A. Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences: Abraham Berman, Robert J. Plemmons*. Number pt. 11 in Classics in Applied Mathematics Series. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1994.
- [4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation*. Prentice Hall, 1989.
- [5] A. Frommer and D. B. Szyld. On asynchronous iterations. *J. of Computational and Applied Mathematics*, 2000.
- [6] F. Gantmacher. *The theory of matrices. 2*. Chelsea Publishing Series. AMS Chelsea Pub, 2000.
- [7] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, 1997.
- [8] E. Hestenes, Magnus R.; Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6).
- [9] D. Hong. D-iteration: application to differential equations. *arXiv*, <http://arxiv.org/abs/1204.1423>, March 2012.
- [10] D. Hong. D-iteration: Evaluation of a dynamic partition strategy. *Proc. of AHPCN 2012*, June 2012.
- [11] D. Hong. D-iteration method or how to improve gauss-seidel method. *arXiv*, <http://arxiv.org/abs/1202.1163>, February 2012.
- [12] D. Hong. Optimized on-line computation of pagerank algorithm. <http://arxiv.org/abs/1202.6158>, 2012.
- [13] D. Hong. Revisiting the d-iteration method: from theoretical to practical computation cost. *arXiv*, <http://arxiv.org/abs/1203.6030>, March 2012.
- [14] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.
- [15] G. Stewart. *Matrix Algorithms, Volume 1 to 5*. SIAM, Philadelphia, 2002.
- [16] R. Varga. *Matrix Iterative Analysis*. Springer Series in Computational Mathematics. Springer, 2009.